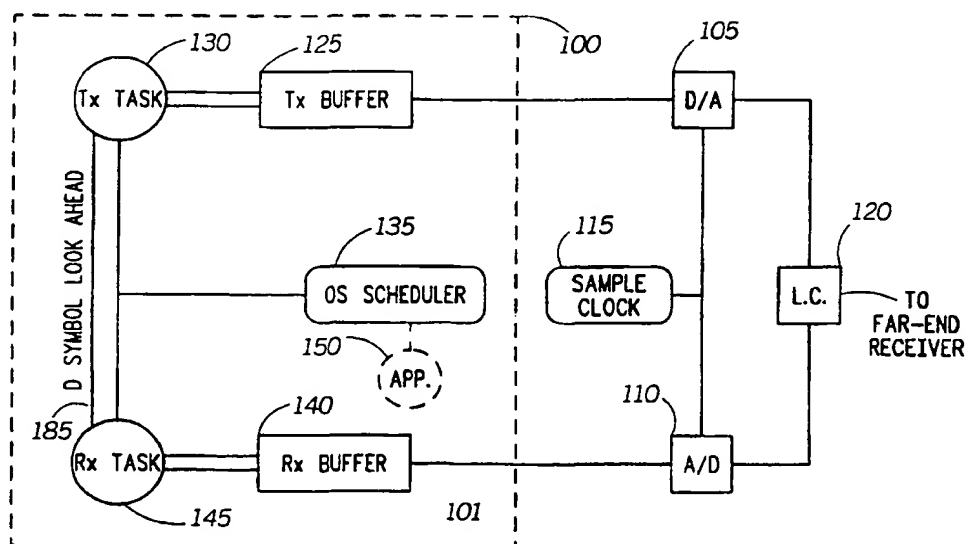




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : H04L 12/56, H04M 11/06		A1	(11) International Publication Number: WO 00/46962
			(43) International Publication Date: 10 August 2000 (10.08.00)
(21) International Application Number: PCT/US00/02673 (22) International Filing Date: 2 February 2000 (02.02.00) (30) Priority Data: 09/243,745 3 February 1999 (03.02.99) US (71) Applicant: MOTOROLA INC. [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US). (72) Inventors: YANG, Jian; 43 Colonial Drive, Mansfield, MA 02048 (US). KRISHNAN, Venkatraman, Gopal; 3 Rockwell Drive, Shrewsbury, MA 01545 (US). (74) Agents: WOOD, J., Ray et al.; Motorola Inc., Intellectual Property Dept., 1303 East Algonquin Road, Schaumburg, IL 60196 (US).		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	

(54) Title: METHOD AND APPARATUS FOR TOLERATING SCHEDULING LATENCY IN HIGH-SPEED MODEMS



(57) Abstract

The preferred embodiment of the present invention provides a method and apparatus for dealing with scheduling latency in high-speed modems (100) implemented on host processors (101). With the present invention, high scheduling latency of the host operating system (135) is tolerated without incurring large end-to-end transmission delay. In addition, time alignment between downstream and upstream signals (175, 180) is achieved even though the scheduling delay may be large and varying.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

METHOD AND APPARATUS FOR TOLERATING SCHEDULING LATENCY IN HIGH-SPEED MODEMS

5

Field of the invention

The present invention relates generally to a method and apparatus for tolerating scheduling latency and achieving time alignment for transmit and receive signals in high-speed modems implemented on host
10 processors.

Background of the Invention

With host processors in personal computers becoming more and more powerful, it becomes feasible to implement high-speed modems
15 such as asymmetrical digital subscriber line (ADSL) modems in software. Computation complexity or millions-of-instructions-per-second, however, are not the only challenges for a successful implementation of high-speed modems. The digital signal processor (DSP) functions for the high-speed modems require real-time execution, e.g., the DSP function must be
20 executed to generate additional transmit samples to be queued to the transmit (TX) buffer before it becomes empty and the DSP function must be executed to process received samples before the receive (RX) buffer becomes full. The host processor, however, may be running many other tasks concurrently such that its operating system may not be able to
25 guarantee the timely execution of the real-time DSP functions. In other words, the scheduling latency, as defined as the delay of actual execution of the task from the time when the request is generated, is very large. Clearly, if the latency exceeds the amount of time it takes to transmit the remaining samples in the TX buffer, a TX buffer underrun problem will

arise. Likewise, if the RX task does not get executed by the time the RX buffer is completely filled, a RX buffer overrun problem will arise.

In addition, the large response time may cause problems for meeting certain requirements. For example, the ADSL standards
5 (G.992.x) require time alignment between upstream and downstream signals, i.e., certain upstream signal must appear at the line at about the same time its corresponding downstream signal is being transmitted by the central site modem. The straightforward approach of starting to send the upstream signal as soon as the corresponding downstream signal is
10 received would fail if the response delay is large.

Thus, there exists a need to provide a method and apparatus that deals with the large response delay caused from the transmit buffer underrun and/or receive buffer overrun, and also resolves its corresponding time alignment problems.

15

Brief Description of the Drawings

A preferred embodiment of the invention is now described, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 illustrates a block diagram of a host processor based ADSL
20 modem according to the preferred embodiment of the present invention;

FIG. 2 illustrates a block diagram of a receive buffer overrun prevention technique according to the preferred embodiment of the present invention;

FIG. 3 illustrates a block diagram of transmit buffer handling
25 according to the preferred embodiment of the present invention;

FIG. 4 illustrates a timing diagram of a downstream and upstream signal time alignment according to the preferred embodiment of the present invention; and

FIG. 5 illustrates a block diagram of a storage medium having stored thereon a set of instructions, which when loaded into a microprocessor, causes the microprocessor to perform a variety of function according to the preferred embodiment of the present invention.

5

Detailed Description of the Preferred Embodiment

The preferred embodiment of the present invention provides a method and apparatus for dealing with scheduling latency in high-speed modems implemented on host processors. With the present invention, high scheduling latency of the host operating system (OS) is tolerated without incurring large end-to-end transmission delay. In addition, time alignment between downstream and upstream signals is achieved even though the scheduling delay may be large and varying.

FIG. 1 illustrates a conceptual block diagram of a high-speed modem 100 implemented on a host processor 101 according to the preferred embodiment of the present invention. The hardware components of the modem 100 consist of a digital-to-analog (D/A) converter 105 and an analog-to-digital (A/D) converter 110, which are driven by a sample clock 115 generator, and a line coupler 120, which typically consists of analog filters, amplifiers and a hybrid. The software components of the modem 100 consist of a TX buffer 125, a TX DSP task 130, a host operating system (OS) task scheduler 135, a RX buffer 140 and a RX DSP task 145.

The TX buffer 125 stores transmit data samples generated by the TX DSP task 130 before the data samples are sent to the D/A converter 105. The RX buffer 140 stores the A/D converter 110 data samples to be processed by the RX DSP task 145. Both the TX and RX buffers 125, 140 are implemented in the host processor's memory. The TX and RX DSP

tasks 130, 145 are executed by the host processor 101 and are scheduled by the host OS task scheduler 135.

A request is generated to the host OS task scheduler 135, typically in the form of an interrupt, either periodically or when the number of
5 samples in the TX buffer 125 is below a threshold for an execution of the TX DSP task 130. Similarly, a request for the RX task 145 is generated either periodically or when the number of samples in the RX buffer 140 is above a threshold.

In a host processor environment, although the worst case
10 scheduling latency can be very large, the typical case is usually one or two orders of magnitude smaller. The preferred embodiment of the present invention utilizes this fact to reduce the impact of the scheduling latency.

The RX buffer 140 overrun problem is minimized by keeping the RX buffer 140 large. Preferably, the RX buffer 140 is large enough to handle
15 worst-case interrupt latency other than unusual cases when the RX DSP task 145 is significantly delayed. As shown in FIG. 2, each time the RX DSP task 145 is executed it processes the entire RX buffer 140 and delivers decoded data to an upper layer application 150. Since the upper layer application 150 that uses the received decoded data typically has a
20 lower scheduling priority than the RX DSP task 145, the RX DSP task 145 scheduling latency does not substantially increase the delay as seen by the upper layer application 150 as long as the amount of time it takes to complete the RX DSP task 145 is not excessive. The RX DSP task 145 can only be blocked by a higher priority level task that would also block the
25 execution of the upper layer application 150.

In the unusual case when the RX DSP task 145 is significantly delayed, the amount of samples in the RX buffer 140 may become excessive to process. In the preferred embodiment of the present invention, the number of samples in the RX buffer 140 is monitored 155.

If the number of samples exceeds a certain threshold at the time the RX DSP task 145 is executed, some of the samples in the RX buffer 140 are discarded/purged to reduce the amount of processing needed 160.

Discarding some of the samples may cause a glitch in the received data stream. In most cases, however, such an infrequent glitch is smoothed out by a higher layer re-transmission protocol or is ignored.

With respect to the TX buffer 125 underrun problem, it is undesirable to keep a lot of samples in the TX buffer 125, as it increases the end-to-end transmission delay. In the preferred embodiment of the present invention, it is not desirable to avoid the TX buffer underrun completely. As shown in FIG. 3, each time the TX DSP task 130 is executed it only generates enough TX data samples such that the number of data samples in the TX buffer 125 reaches a certain threshold. When the TX buffer 125 is full, the TX DSP task 130 is not delayed.

In addition, pad samples 165 are appended to the end of the TX buffer 125. Pad samples 165 are patterns that are least destructive to the far-end-receiver (e.g., for ADSL, this could be a synchronization symbol). As a result, the TX buffer underrun results in the pad samples 165 being sent, which merely causes a recoverable glitch in the far-end-receiver (not shown). The TX buffer threshold is designed such that TX buffer underrun does not occur for typical scheduling delays, i.e., TX buffer underrun does not occur often.

Clearly, it is critical to set the TX buffer threshold to best balance the delay and the occurrence of the glitch. As such, the preferred embodiment of the present invention adaptively adjusts the threshold setting based on the host processor's speed and loading. In order to adaptively adjust the threshold setting based on the host processor's speed and loading, the depth of the TX buffer 125 is monitored 170 each time the TX DSP task 130 is executed. Every time a TX DSP task 130 is

executed, the number of data samples stored in the TX buffer 125 is recorded. The depth of the TX buffer 125 is used as metrics for TX buffer threshold adjustments. Several algorithms can be used to accomplish the TX buffer threshold adjustments. The basic idea of the algorithms is to
5 increase the thresholds when the depth of the TX buffer 125 is too small and vice versa. As an example, a user can record the number of TX buffer underruns during a time interval of ten (10) seconds. If the number of TX buffer underruns exceeds a predetermined value or is above a specific design goal (e.g., the user only wants an underrun to occur three (3)
10 times during the ten (10) second time interval), the TX buffer threshold is increased. In order to increase the TX buffer threshold, the TX buffer 125 queues extra data samples from the subsequent executed TX DSP tasks
130.

A further example is to calculate the average of the depth of the TX
15 buffer 125 during a specified time interval (e.g., a one (1) second time interval). If the calculated average is below a specified design goal, the TX buffer threshold is increased.

In addition to buffer underrun/overflow problem, now resolved by the preferred embodiment of the present invention above, scheduling
20 latency also causes difficulty for achieving time alignment between transmit and receive signals as required by ADSL standards. As shown in FIG. 4, there is a state transition in the downstream signal 175 from segment A to segment B and in the upstream signal 180 from segment a
to segment b. For example, if using the ADSL standard, segment A could
25 be the C-REVER3 signal defined in G.992.2 and segment B could be the C-SEGUE1 signal, segment a could be the R-REVERB2 signal and segment b could be the R-SEGUE1 signal. Looking from the telephone line, the a/b transition in the upstream signal 180 should occur simultaneously with the A/B transition in the downstream signal 175. Due to the scheduling

latency and the relatively large delay in the RX and TX buffers 125, 140, it is not feasible for the client modem to start generating TX data samples for segment b upon detecting the A/B transition in the downstream signal 175.

5 Referring back to FIG. 1, the sample clock 115 is locked between the D/A and A/D converters 105, 110, i.e., the two sample clocks 115 are derived from the same source. In addition, the scheduling of the TX and RX tasks 130, 145 is also locked, i.e., either the TX or RX interrupt is used to schedule both TX and RX DSP tasks 130, 145. As such, the sum of the
10 local transmit delay and the local receive delay (i.e., the local echo path delay) is constant in the sense that it does not vary with the scheduling delay. This is because a delayed interrupt service causes a shortening of transmit delay as the data samples in the TX buffer 125 are being taken to the D/A converter 105 at the constant sampling rate. This shortening of
15 transmit delay, however, is matched exactly by a lengthening of receive delay as samples from the A/D converter 110 are being fed into the RX buffer 140 at exactly the same rate. It should be noted that the constant local echo path delay is in terms of the product of number of samples and the corresponding sampling interval rather than simply number of
20 samples. There is a need for such a distinction as the TX and RX sampling rates may be different. In particular, for ADSL modems, the constant local echo path delay can be represented in terms of symbols, as the ADSL symbol rate is the same for downstream and upstream.

In addition to locking the local echo path delay, the preferred
25 embodiment of the present invention also drives the TX DSP task 130 based on a "look-ahead" scheme 185. In other words, the preferred embodiment of the present invention anticipates the next transmit state transition and queues the appropriate transmit signal to the TX buffer 125 before the corresponding RX signal is received/processed. In particular, if

"D" is the constant sum of delays in symbols, a "D symbol look-ahead" scheme 185 is required. Referring back to the previous example, the TX DSP task 130 starts generating samples for segment b "D" symbols earlier than the A/B transition is detected in the downstream signal 175.

5 The D symbol look-ahead scheme in conjunction with locking the TX and RX sample clock/processing guarantees the alignment between the downstream and upstream signals 175, 180. To verify this, consider the case when transmit delay is zero and receive delay is "D" symbols. The "D symbol look-ahead" scheme 185 described above clearly ensures the
10 alignment in this particular case. In the more general case, when transmit delay is $d > 0$, receive delay must then be $D - d$, the TX DSP task 130 queues the first sample of segment b to the TX buffer 125 $D - (D - d) = d$ symbols before the A/B transition occurs at the line. Since the TX delay in this case is d , the a/b transition would occur at the same time as the A/B
15 transition.

In order for the "D symbol look-ahead" scheme 185 to be feasible, the TX DSP task 130 must know what TX samples need to be generated "D" symbols before its corresponding RX signal is processed by the RX DSP task 145. In the example above, the RX DSP task 145 needs to know
20 when the A/B transition in the downstream signal 175 is going to occur "D" symbols before it finishes processing the last symbol of segment A. This is needed for signaling the TX DSP task 130 to complete sending upstream signal segment a and start sending segment b. This problem is dealt with in one of three ways.

25 Firstly, an earlier state transition in the RX signal is used to predict the next state transition. Sometimes, it is a known fixed number of symbols between the previous state transition and the upcoming state transition. By counting the number of received symbols, the occurrence of the upcoming state transition is accurately predicted. Therefore, the

transmit task is informed of which transmit samples need to be generated by using a previous state transition in the receive signal to predict the upcoming state transition, wherein the receive signal has a known fixed number of symbols between the previous state transition and the
5 upcoming state transition, and by counting a number of received symbols, an occurrence of the upcoming state transition is accurately predicted.

Secondly, in some cases the RX DSP task 145 may be able to generate early decisions. An example is the case where the modem 100 needs to start sending segment b upon detecting the presence of signal
10 segment A. In such a case, the RX DSP task 145 needs to be able to detect signal A before observing the last symbol of the signal segment. This is often possible especially if the signal segment A is a tonal signal. Therefore, the transmit task is informed of which transmit samples need to be generated by detecting a certain receive signal; determining a
15 presence of the certain receive signal at least D symbols before the certain receive signal terminates; and sending a response to the certain receive signal upon determining its presence.

Thirdly, in some cases the state transition from A to B is known (can be predicted), but the content of signal b depends on the reception
20 of the signal A. As an example, segment A is used by the modem 100 to estimate certain channel parameters, and the upstream signal 180 sent in segment b depends on the result of the estimation. In such cases, the modem 100 in the present invention does not use the "D" symbols at the end of the segment A for the estimation, i.e., the channel estimation
25 terminates D symbols before the end of the segment A, and the result of the prematurely terminated estimation are used for generating segment b in the upstream signal. Therefore, the transmit task is informed of which transmit samples need to be generated by estimating certain channel parameters; prematurely terminating the step of estimating D symbols

before an end of its corresponding receive signal; and using the prematurely terminated estimation for generating the upcoming state transition.

Thus, the preferred embodiment of the present invention achieves
5 time alignment via locking the local echo path delay and driving the transmit state transition based on a "look ahead" scheme. As described above, the sum of the local transmit delay and the local receive delay are fixed even though the interrupt latency varies.

Assuming that "D" is the constant local echo path delay (in unit of
10 symbols). A "D symbol look ahead" scheme is then implemented. In other words, the transmit signal is queued when the receiver is processing the D'th symbol before the corresponding far end signal. Such a configuration ensures that the transmit signal appears at the phone line at the right instant.

15 While the invention has been described in conjunction with a specific embodiment thereof, additional advantages and modifications will readily occur to those skilled in the art. For example, as shown in FIG. 5, the preferred embodiment of the present invention can be embodied in a storage medium 185 having stored thereon a set of
20 instructions which, when loaded into a microprocessor 190, causes the microprocessor 190 to perform a variety of functions (some of which are described below), in accordance with the preferred embodiment of the present invention.

The storage medium 185 could have stored thereon a set of
25 instructions that would cause the microprocessor 190 to execute a transmit task; generate data samples such that a number of data samples in a transmit buffer reaches a predefined buffer threshold; append pad samples to an end of the transmit buffer; and transmit the pad samples to a far-end-receiver in an event of a transmit buffer

underrun, wherein the transmit buffer underrun occurs when the data samples in the transmit buffer are depleted.

The storage medium 185 could have stored thereon a set of instructions that would also cause the microprocessor 190 to execute a receive task; processing data samples from a receive buffer; monitor the
5 number of data samples in the receive buffer; and if the number of data samples in the receive buffer exceeds a threshold at a time the receive task is executed, purge a portion of the data samples in the receive buffer.

10 The storage medium 185 could have stored thereon a set of instructions that would cause the microprocessor 190 to transport transmit data samples from a transmit buffer to a digital-to-analog converter at a transmit sampling rate; transport receive data samples from an analog-to-digital converter to a receive buffer at a receive sampling rate; lock a
15 sample clock between the digital-to-analog converter and the analog-to-digital converter such that the digital-to-analog converter and the analog-to-digital converter are derived from a common clock source; and lock a scheduler between a transmit task and a receive task such that a sum of a local transmit delay and a local receive delay is constant.

20 The invention, in its broader aspects, is therefore not limited to the specific details, representative apparatus, and illustrative examples shown and described. Various alterations, modifications and variations will be apparent to those skilled in the art in light of the foregoing description. Thus, it should be understood that the invention is not
25 limited by the foregoing description, but embraces all such alterations, modifications and variations in accordance with the spirit and scope of the appended claims.

Claims

We claim:

- 5 1. A method for tolerating scheduling latency in a high-speed modem implemented on a host processor comprising:
 - executing a transmit task;
 - generating data samples such that a number of data
 - 10 samples in a transmit buffer reaches a predefined buffer threshold;
 - appending pad samples to an end of the transmit buffer; and
 - 20 sending the pad samples to a far-end-receiver in an event of a transmit buffer underrun, wherein the transmit buffer underrun occurs when the data samples in the transmit buffer are depleted.
- 15 2. The method according to claim 1 wherein the predefined buffer threshold is adaptively adjusted based on the host processor's speed and loading by monitoring a depth of the transmit buffer each time the transmit task is executed, wherein a history of
- 20 depths of the transmit buffer are used as metrics for the predetermined threshold adjustments.
3. The method according to claim 3 wherein the predetermined buffer threshold is increased when the depth of the transmit buffer is too small.
- 25 4. The method according to claim 4 wherein the predetermined buffer threshold is increased by having the transmit buffer queue extra data samples from subsequent executed transmit tasks.

5. The method according to claim 3 further comprising:
calculating an average of the depth of the transmit buffer
during a specified time interval; and
if the average calculated is below a predetermined value,
5 increasing the predetermined buffer threshold.
6. A method for tolerating scheduling latency in a high-speed modem
implemented on a host processor comprising:
10 executing a receive task;
processing data samples from a receive buffer ;
monitoring the number of data samples in the receive buffer;
and
if the number of data samples in the receive buffer exceeds
15 a threshold at a time the receive task is executed, purging a portion
of the data samples in the receive buffer.
7. An apparatus for tolerating scheduling latency in high-speed
20 modems implemented on host processors comprising, in hardware:
a digital-to-analog converter;
an analog-to-digital converter;
a sample clock locked between the digital-to-analog
converter and the analog-to-digital converter in order to operate
25 the digital-to-analog converter and the analog-to-digital converter
from a common clock source; and
a line coupler, coupled to the digital-to-analog converter and
to the analog-to-digital converter.

8. The apparatus according to claim 7 further comprising, in software:
a transmit buffer, coupled to the digital to analog converter;
a transmit task, coupled to the transmit buffer;
a receive buffer coupled to the analog-to-digital converter;
5 a receive task, coupled to the receive buffer; and
a host operating system scheduler, coupled to the transmit task and the receive task in order to lock any scheduling of the transmit task and the receive task.
- 10 9. A storage medium having stored thereon a set of instructions which, when loaded into a microprocessor, causes the microprocessor to perform the following functions:
executing a transmit task;
generating data samples such that a number of data
15 samples in a transmit buffer reaches a predefined buffer threshold; and
appending pad samples to an end of the transmit buffer; and
sending the pad samples to a far-end-receiver in an event of a transmit buffer underrun, wherein the transmit buffer underrun
20 occurs when the data samples in the transmit buffer are depleted.
10. A storage medium having stored thereon a set of instructions which, when loaded into a microprocessor, causes the microprocessor to perform the following functions:
25 executing a receive task;
processing data samples from a receive buffer ;
monitoring the number of data samples in the receive buffer;

if the number of data samples in the receive buffer exceeds a threshold at a time the receive task is executed, purging a portion of the data samples in the receive buffer.

- 5 11. A storage medium having stored thereon a set of instructions which, when loaded into a microprocessor, causes the microprocessor to perform the following functions:

transporting transmit data samples from a transmit buffer to a digital-to-analog converter at a transmit sampling rate;

10 transporting receive data samples from an analog-to-digital converter to a receive buffer at a receive sampling rate;

locking a sample clock between the digital-to-analog converter and the analog-to-digital converter such that the digital-to-analog converter and the analog-to-digital converter are derived
15 from a common clock source; and

locking a scheduler between a transmit task and a receive task such that a sum of a local transmit delay and a local receive delay is constant.

1/2

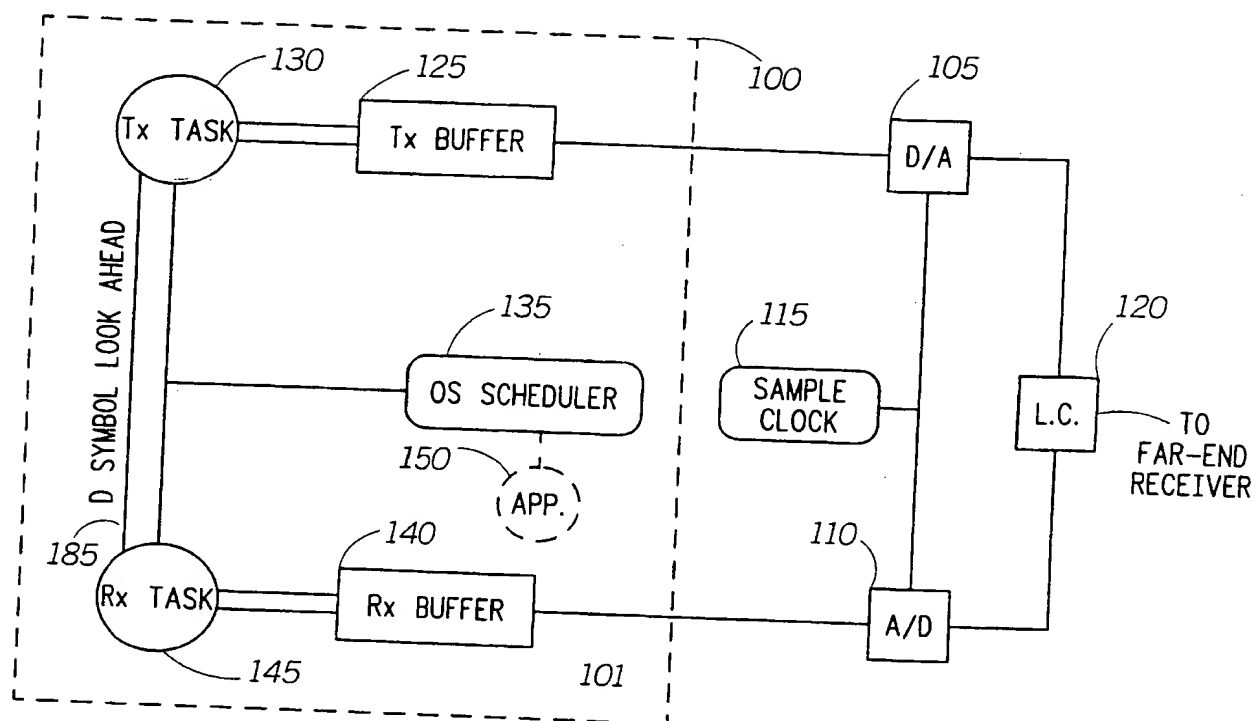


FIG. 1

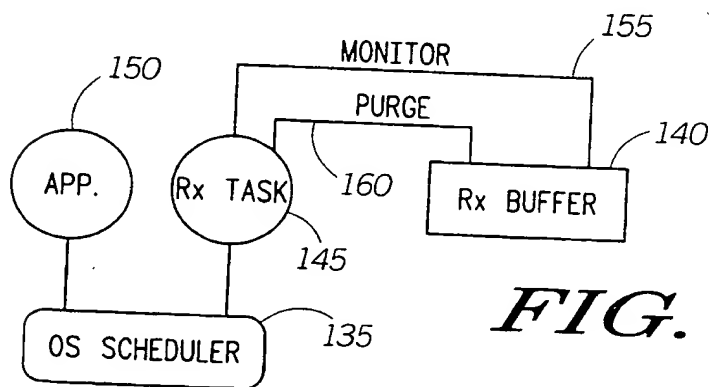


FIG. 2

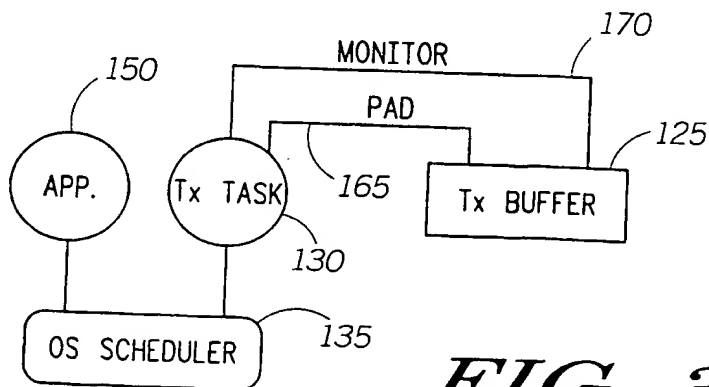
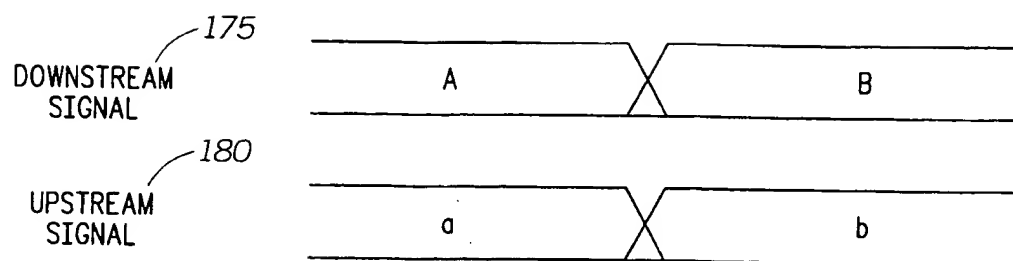
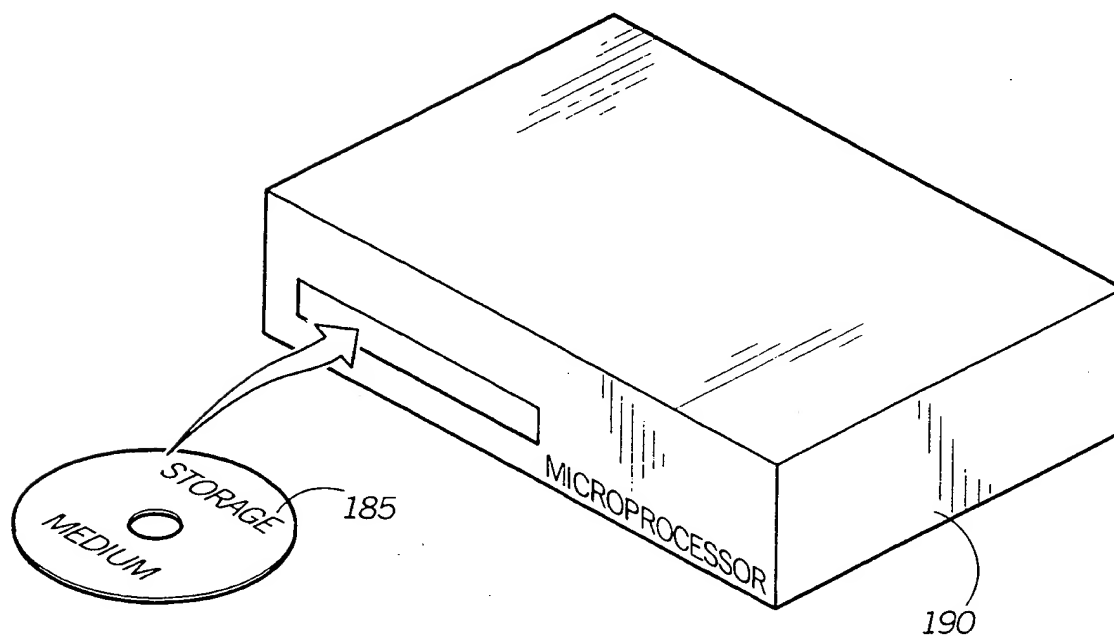


FIG. 3

2/2

*FIG. 4**FIG. 5*

INTERNATIONAL SEARCH REPORT

Internat. Application No.

PCT/US 00/02673

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/56 H04M11/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L H04M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 762 655 A (PC TEL INC) 12 March 1997 (1997-03-12) abstract page 1, line 55 -page 2, line 30	1,9
Y	page 7, line 43 - line 57 figures 1A,1B	2-5
X	US 5 822 540 A (CALDARA STEPHEN A ET AL) 13 October 1998 (1998-10-13) abstract column 1, line 52 -column 2, line 31 figure 2	6,10

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *&* document member of the same patent family

Date of the actual completion of the international search

5 June 2000

Date of mailing of the international search report

13/06/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Toumpoulidis, T

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/02673

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 799 064 A (HOANG MINH ET AL) 25 August 1998 (1998-08-25)	7,8
A	abstract column 2, line 66 -column 4, line 19 column 4, line 42 -column 7, line 16 column 7, line 50 -column 8, line 7 figure 3	11
Y	US 5 384 780 A (LOMP MARTIN G ET AL) 24 January 1995 (1995-01-24) abstract column 1, line 58 -column 2, line 8 column 4, line 29 - line 56 -----	2-5

Information on patent family members

PCT/US 00/02673

Form PCT/ISA/210 (patent family annex) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internat. Application No

PCT/US 00/02673

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5384780 A		JP 7509115 T WO 9427388 A	05-10-1995 24-11-1994
<hr/>			